View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Radboud Repository

Radboud Repository

Radboud University Nijmegen

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a publisher's version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/29099

Please be advised that this information was generated on 2017-12-05 and may be subject to change.

Anal. Chem. **1996**, *68*, 2419–2425

A Genetic Algorithm for Model-Free X-ray Fluorescence Analysis of Thin Films

Adrie D. Dane,*,† Patrick A. M. Timmermans,† Hans A. van Sprang,‡ and Lutgarde M. C. Buydens†

Katholieke Universiteit Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, and Philips Research Laboratories, Professor Holstlaan 4, 5656 AA Eindhoven, The Netherlands

Single-layer films and multiple-layer films can be quantitatively analyzed using X-ray fluorescence. Due to the absence of adequate standards, most methods are based on the calculation of theoretical X-ray fluorescence intensities from fundamental parameter methods. These methods require as initial estimates the exact qualitative sample structure and accurate starting values for both concentrations and layer thicknesses. This paper proposes a fundamental parameter method that uses a genetic algorithm as an optimization procedure. The relaxation of the requirements on the description of the sample due to this robust optimization is discussed. Preliminary results are presented indicating possible applications as well as areas for further research.

Both single-layer and multiple-layer films are important in many industrial and scientific applications. Examples of these materials are multiple thin layers of metal on silicon wafers in the electronics industry and multiple coatings of sheet metals on suitable substrates for corrosion protection. These materials can be quantitatively analyzed by X-ray fluorescence. Advantages of X-ray fluorescence for these applications are that it is nondestructive and that a wide range of elements can be quantified. Furthermore, it is possible to determine concentrations and layer thicknesses simultaneously.

Because of the absence of adequate standards, layered materials are often analyzed using fundamental parameter methods.[1–5] These methods are based on the calculation of theoretical X-ray fluorescence intensities for a given composition under a given set of experimental conditions. In a conventional fundamental parameter method, a sample composition is iteratively updated until calculated and measured intensities are sufficiently close. The final concentrations and layer thicknesses used are assumed to represent the actual composition.

A drawback of optimization algorithms currently applied in fundamental parameter methods is that initial sample estimates have to be rather accurate, where the actually required accuracy may depend on the type of sample and the optimization procedure used. The estimates have to obey the following conditions: the number of layers must be known, the qualitative composition of the different layers must be known, and initial estimates of the quantitative composition must be provided. This paper explores to what extent less adequate initial estimates lead to good quantitative results. Genetic algorithms represent a class of methods potentially well suited for this purpose. For this reason, we investigated a genetic algorithm as part of an overall fundamental parameter procedure in this work.

So far, our method has been tested only on simulated data. For relatively simple samples, good quantitative results are obtained if the number of layers is kept constant. Furthermore, this procedure gives promising results when the input model assumes more layers than present. More complicated sample structures are still under study.

FUNDAMENTAL PARAMETER METHOD

The aim of X-ray fluorescence analysis is to find the correct set of quantitative sample parameters from a set of measured X-ray fluorescence intensities. For bulk materials, these sample parameters (concentrations) are normally obtained from a calibration using a set of calibration samples. This method is generally not applicable for multilayer films since calibration samples are very difficult to prepare, since both the composition and the layer thicknesses have to be varied and have to be known accurately. Therefore, the only practical analytical method is based on the calculation of theoretical intensities from physical parameters.

Theoretical X-ray intensities can be calculated from physical principles using so-called fundamental parameter equations. These fundamental parameter equations for bulk materials were first published in 1968 by Criss and Birks.[1] The equations were extended to multilayer materials in the mid-1980s by Mantler.[6,7] However, Mantler's expressions for secondary fluorescence contained an angular integral which had to be calculated numerically. Analytical expressions in terms of the exponential integral were presented by de Boer[2] and can be evaluated using fast converging series.[8,9] A detailed description is outside the scope of this paper. The reader is referred to the citations given.

The combination of the calculation of theoretical X-ray intensities and an optimization algorithm is called a fundamental parameter method. A fundamental parameter method for the analysis of multilayer samples usually consists of the following steps: assuming an approximate composition for the unknown specimen, calculating the fluorescence intensities from equations

† Katholieke Universiteit Nijmegen.

‡ Philips Research Laboratories.

(1) Criss, J. W.; Birks, L. S. *Anal. Chem.* **1968**, *40*, 1080–1086.
(2) de Boer, D. K. G. *X-Ray Spectrom.* **1990**, *19*, 145–154.
(3) de Boer, D. K. G.; Borstrok, J. J. M.; Leenaerts, A. J. G.; van Sprang, H. A. *X-Ray Spectrom.* **1993**, *22*, 1–6.
(4) Tertian, R.; Claisse, F. *Principles of quantitative X-Ray fluorescence analysis*; Heyden: London, 1982.
(5) Dane, A. D. In *Chemometrics: exploring and exploiting chemical information*; Buydens, L. M. C., Melssen, W. J., Eds.; Katholieke Universiteit Nijmegen: Nijmegen, The Netherlands, 1994; Chapter 14, pp 227–242.
(6) Mantler, M. *Advances in X-ray Analysis*; Plenum: New York, 1984; pp 433–440.
(7) Mantler, M. *Anal. Chim. Acta* **1986**, *188*, 25–35.
(8) Cody, W. J.; Thacher, H. C., *Jr. Math. Comput.* **1968**, *22*, 641–649.
(9) Cody, W. J.; Thacher, H. C., *Jr. Math. Comput.* **1969**, *23*, 289–303.

S0003-2700(95)01152-8 CCC: $12.00 © 1996 American Chemical Society

involving fundamental parameters (densities, attenuation coefficients, etc.), and comparing the calculated intensities with the measured intensities. The composition is iteratively improved until the theoretical and measured intensities are consistent according to an error criterion. Final concentrations used are assumed to represent the actual composition. A fundamental parameter method based on the equations presented by de Boer, a primary spectral distribution calculated from an algorithm,[10,11] and a Gauss–Newton optimization is successfully applied to samples in which the qualitative structure is known.[3] The intensity calculations have been implemented in our program.

Optimization. In most previous fundamental parameter implementations,[1,3,7] some form of steepest descent method[12,13] is used in optimizing the sample model. Steepest descent methods iteratively improve a candidate solution by adaption in the direction of the largest error decrease. For this purpose, derivatives are calculated at the current position. The methods adapt the trial solution in the direction of the largest error decrease at the current position, which causes them to converge to the nearest optimum. If multiple optima exist and no good starting values are given, then these algorithms are likely to converge to a local optimum. To avoid this premature convergence, all knowledge about the problem should be included to obtain good starting values.[12] If there is not much knowledge, an alternative is to take different starting values, which can be chosen either by performing an experimental design in the search space or, for larger problems, at random.[12] In these methods, all starting positions are evaluated, and the optimization is started at one or more points with low errors. The optimal solution found is the final solution with the lowest error. This solution is not necessarily the global optimum. The dependency on good starting values makes steepest descent methods not so robust in complex problems. A very strong property of steepest descent methods is, however, that the position of the final solution is reached with high precision.

Optimization Using Genetic Algorithms. If, in fundamental parameter optimization, the exact sample structure is not supplied, the initial estimates are inaccurate. Here, steepest descent is very likely to fail. Genetic algorithms are far less sensitive to local optima and have been robust in optimizing various complex applications.[14,15] Genetic algorithms differ from more traditional optimization methods in some fundamental ways.[16]

A genetic algorithm[17] uses a search strategy based on biological evolution. It processes a population of candidate solutions (strings) simultaneously instead of a single point. In each iteration, the whole population is replaced by a new population. By analogy to nature, this process is called a generation. The genetic iteration cycle will be described below.

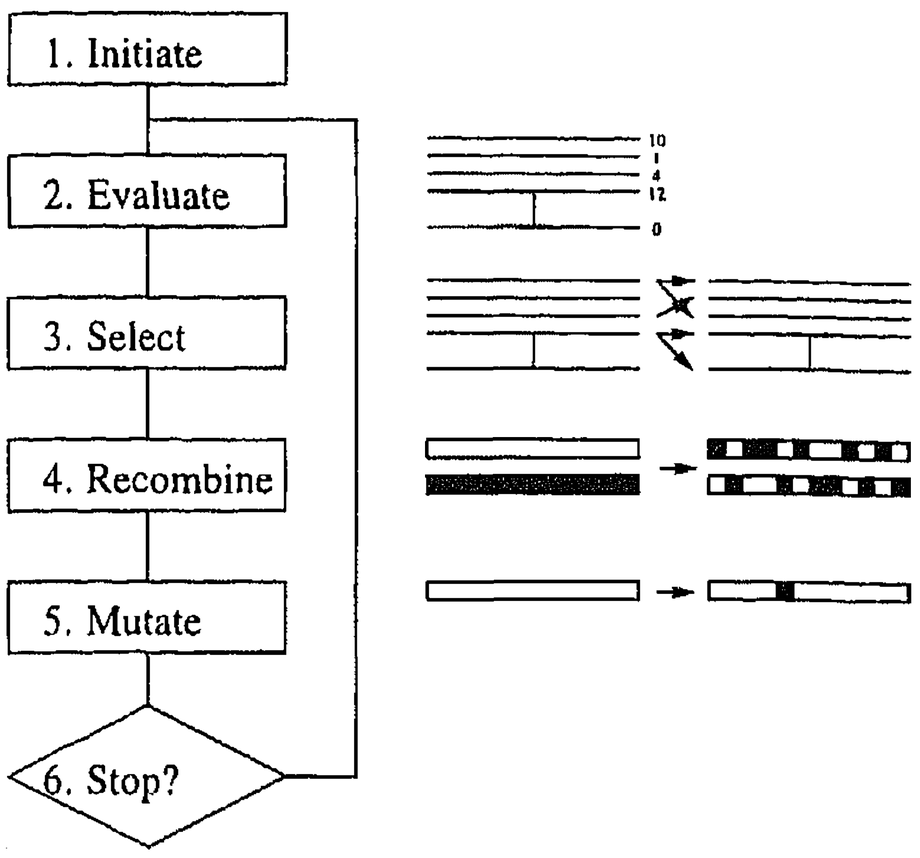

Figure 1. Flow chart of a genetic algorithm.

To work most effectively, genetic algorithms use binary coded versions of the parameter strings.[16] In these so-called bitstrings, each parameter is encoded as a bitfield of B bits. By applying this encoding, the value range of the unknown parameter is subdivided into 2^B levels. Larger values of B amount to a more finely meshed search grid. Our experience is that, once the value of B is large enough for the required accuracy, additional bits do not have a noticeable effect on the performance. The coding ensures that only valid parameter values can be represented/generated.

In contrast to other methods, genetic algorithms do not use derivatives or other auxiliary knowledge; genetic algorithms use result information. Furthermore, genetic algorithms use probabilistic instead of deterministic transition rules. This makes their search precision very poor. Genetic algorithms are robust in locating the neighborhood of the global optimum, but the exact location is rarely obtained. To obtain higher quality solutions, a steepest descent method as second optimization step is often used.

For each step in a genetic algorithm, there are different operators from which to choose. In each operator, one or more control parameters must be set. This configuration freedom makes genetic algorithms not very user-friendly. On the other hand, it makes genetic algorithms very flexible and tunable to perform well in solving different types of problems.[14,15] Following the flow chart depicted in Figure 1, the steps in a genetic algorithm will be discussed briefly. Furthermore, a more detailed description of the genetic operators used in our implementation will be given. The choice for these operators is based largely on literature and experience. The values for the corresponding control parameters are supplied in the Experimental Section. The settings are based on experience and some preliminary experiments. For more detailed information on this subject, the reader is referred to refs 14–16 and 18.

(1) Initiate. In this step, the initial population is generated. In our case—as in most genetic algorithms—this is done by assigning each bit in the population a random value.

(2) Evaluate. All strings are evaluated and receive a quality value (*fitness*). This fitness is small when the *error* is large, and vice versa.

In our case, the evaluation consists of the calculating theoretical intensities and comparing them to the measured intensities by calculating an error. The fitness is obtained by taking error^{-1}.

(10) Pella, P. A.; Feng, L.; Small, J. A. *X-Ray Spectrom.* **1985**, *14*, 125–135.
(11) Pella, P. A.; Feng, L.; Small, J. A. *X-Ray Spectrom.* **1991**, *20*, 109–110.
(12) Draper, N. R.; Smith, H. *Applied regression analysis*, 2nd ed.; Wiley series in probability and mathematical statistics; Wiley: New York, 1981.
(13) Press, W. H.; Flannery, B. P.; Teukolski, S. A.; Vetterling, W. T. *Numerical Recipes in C, The Art of Scientific Computing*; Cambridge University Press: Cambridge, U.K., 1988.
(14) Lucasius, C. B.; Kateman, G. *Chemom. Intell. Lab. Syst.* **1993**, *19*, 1–33.
(15) Lucasius, C. B.; Kateman, G. *Chemom. Intell. Lab. Syst.* **1994**, *25*, 99–146.
(16) Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Reading, MA, 1989.
(17) Holland, J. H. *Adaption in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, 1975; Revised: MIT Press, Cambridge, MA, 1992.
(18) Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Artificial Intelligence Series; Springer-Verlag: Berlin, 1992.

(3) Select. From the population, strings are selected with rates proportional to their fitnesses to yield an equally sized new population. The biological counterpart of selection is *survival of the fittest.* Selection yields a population with a higher average quality than the old population.

In our genetic algorithm, we use two selection methods: elitist selection and threshold selection.[16] In elitist selection, the best F_{elitist} fraction of the population is copied to the new population. Recombination and mutation do not operate on this portion of the new population. Elitism guarantees that the unmodified best solution always will be present in the next generation.

The rest of the new population is obtained by threshold selection. In threshold selection, strings from the best $F_{\text{threshold}}$ fraction of the old population are chosen at random and copied to the new population until the new population is the same size as the old population.

(4) Recombine. In this step, each string in the population (*or, as in our case, each string in the non-elitist-selected part of the population*) is paired with another string. Recombination (also called crossover) is carried out on each string pair with probability P_r. In this recombination, parts of two solutions are exchanged. If two high-quality regions are combined, this operator will yield a higher quality solution. Both the selection pressure and the recombination, together with a good representation, guide the convergence of a genetic algorithm.

In our implementation, we apply uniform binary crossover.[19] A uniform crossover exchanges each bit position between the two strings, with probability P_s.

(5) Mutate. Selection can lead to a population with no more strings containing high-quality regions at different positions. If this is the case, the probability that recombination leads to higher quality solutions will become too low, and improvement of the best solution within the population will stop. To prevent this premature convergence, the mutation operator is introduced. Mutation can reintroduce information lost during selection by changing a small part of the string at random. As mutation operator, we used uniform binary mutation. In uniform binary mutation, each bit in the population (*or, as in our case, each bit in the non-elitist-selected part of the population*) is inverted with probability P_m.[17]

(6) Stop? Steps 2–5 are called a generation, and they are repeated until a certain stop criterion is met. Typical stop criteria in a genetic algorithm run are a predefined maximum number of generations or an error smaller than a predefined value. In our genetic algorithm, a maximum number of generations is used.

APPLICATION OF THE METHOD

To apply a genetic algorithm to a given problem, a good representation must be chosen, and an evaluation function mapping a candidate solution to the fitness must be designed. Both will be described below, followed by a description of the experiments performed.

Representation. The candidate solutions must be represented as strings (vectors) of proposed values for the unknown parameters, in our case thicknesses and concentrations. These vectors are coded, requiring an upper and a lower bound for each parameter. A good representation should use all knowledge available to keep the number of parameters and their value ranges as small as possible, and additional knowledge must easily be incorporated. Furthermore, the generation of physically impossible solutions by the genetic operators must be prevented. The following physical constraints apply to layered samples: (1) layer thicknesses must be equal to or larger than 0, (2) concentrations must lie between 0 and 1, and (3) the sum of the concentrations of all elements present in one layer must be equal to 1. As an additional restriction, we consider that the overall qualitative sample composition could be extracted from the spectrum. This could be defined as an additional constraint: (4) elements for which no intensities can be detected are not present in the sample.

Figure 2A depicts a sample consisting of a 1 μm layer of titanium (20%), arsenic (40%), and silver (40%) on a 30 μm silicon (100%) carrier. Figure 2B–D depicts qualitative descriptions of Figure 2A with varying amounts of qualitative knowledge. Figure 2B is the exact qualitative description of Figure 2A; this represents the knowledge required by currently applied fundamental parameter methods. Figure 2C depicts a qualitative description of Figure 2A, in which the number of layers is known but the distribution of the elements is not. In Figure 2D, both the number of layers and the distribution of the elements are unknown.

The quantitative composition of a layer (e.g., the upper layer in Figure 2A) can be obtained from a number of parameters equal to the number of elements, n, in the description of that layer.

The most obvious string representation of a layer for which n elements have to be optimized would be to describe it by its thickness and $n - 1$ concentrations. The drawback of this representation method is caused by the fact that the concentra-

Ti(0.2)	As(0.4)	Ag(0.4)	↕ 1μm
	Si(1.0)		↕ 30μm

A A layered sample consisting of a 1μm layer of Ti(20%), As(40%), Ag(40%) on 30μm Si(100%). The quantitative sample structure, i.e. the number of layers, their individual quantitative compositions and their thicknesses, determines the line intensities measured in *X*-ray fluorescence spectrometry.

Ti_1	As_1	Ag_1	↕ d_1
	Si_2		↕ d_2

B Exact qualitative composition of Figure 2A.

Ti_1	As_1	Ag_1	Si_1	↕ d_1
Ti_2	As_2	Ag_2	Si_2	↕ d_2

C Qualitative composition of Figure 2A in which the distribution of the elements is not known.

Ti_1	As_1	Ag_1	Si_1	↕ d_1
Ti_2	As_2	Ag_2	Si_2	↕ d_2
⋮	⋮	⋮	⋮	
Ti_m	As_m	Ag_m	Si_m	↕ d_m

D Qualitative composition of Figure 2A in which the distribution of the elements and the number of layers is not known.

$0.2_{(Ti_1)}$	$0.4_{(As_1)}$	$0.4_{(Ag_1)}$	$0.0_{(Si_1)}$	$0.0_{(Ti_2)}$	$0.0_{(As_2)}$	$0.0_{(Ag_2)}$	$30_{(Si_2)}$

E String representation of A in which each layer is represented by the products of the fractions and the layer thickness (i.e. elemental thicknesses). A layer thickness is obtained by summing all elemental thicknesses belonging to that layer. A concentration can be calculated by division of the corresponding elemental thickness by the layer thickness.

Figure 2. (A) Sample consisting of a 1 μm layer of Ti (20%), As (40%), and Ag (40%) on 30 μm Si (100%). (B–D) Qualitative descriptions of A with different levels of structural knowledge. (E) String representation of A based on the qualitative description of C.

(19) Syswerda, G. In *Proceedings of the third international conference on genetic algorithms*; Schaffer, J. D., Ed.; Morgan Kaufmann Publishers: San Mateo, CA, 1989.

tions depend on each other, while during optimization they are independently varied. Operators modifying the candidate solutions have to be restricted to exclusively generate solutions in which the concentration sum of the $n - 1$ elements is equal to or less than 1. Otherwise, the last concentration would be negative. To be able to use standard genetic operators without penalty functions, we rejected this method. In our representation, a layer with n elements is represented by the products of elemental concentrations and the layer thickness (in micrometers). We define the parameter values obtained in this way as elemental thicknesses. They represent the part of the layer thickness occupied by the corresponding element. From this representation, a layer thickness can be obtained by summing the products belonging to the layer. An elemental concentration is obtained by dividing the corresponding product by this layer thickness.

For coding purposes, the value ranges of the elemental thicknesses must be defined. In our application, upper and lower parameter bounds are calculated using

$$\text{lower bound} = \text{concentration lower bound} \times \text{thickness upper estimate} \quad (1a)$$

$$\text{upper bound} = \text{concentration upper bound} \times \text{thickness upper estimate} \quad (1b)$$

For our experiments, we wanted to use only physical information and to keep additional structural information to as little as possible. Using the constraints given above, the elemental thickness lower bound is equal to 0, and the elemental thickness upper bound is equal to the thickness upper estimate. This parameter must be supplied by the user. The parameter bounds defined by eq 1 impose the following restrictions upon the underlying concentrations and thicknesses:

Upper Bounds. An upper parameter bound represents the maximum layer thickness for which the corresponding concentration upper bound can be obtained. The sum of the upper bound values for one layer represents the maximum layer thickness that can be obtained.

Lower Bounds. The sum of the lower parameter bounds for an individual layer represents the smallest layer thickness that can be obtained. An individual layer parameter bound represents the minimal layer thickness necessary to obtain the lower concentration bound.

Evaluation Function. For calculating the fitness from the differences between measured and calculated intensities, we use an *elementwise relative error*, which is given by

$$\text{error} = \frac{\sum_{i}^{n\ \text{elements}} \left(\sum_{j}^{n\ \text{lines}_i} \left| \frac{I_{s,ij} - I_{p,ij}}{I_{s,ij}} \right| / n\ \text{lines}_i \right)}{n\ \text{elements}} \quad (2)$$

in which $I_{s,ij}$ represents the simulated intensity of line j belonging to element i, and $I_{p,ij}$ represents the predicted intensity of line j belonging to element i. This error has the following properties: it is equally sensitive to all lines measured for a given element, it is equally sensitive to all elements present, and it is also easily interpreted as a convergence measure for our genetic algorithm.

Experimental Specifications. In all our experiments, the following conditions and settings were used.

(a) X-ray Fluorescence Data. In all experiments, we used simulated data. These simulations were performed by calculating theoretical intensities for a given sample composition. All intensities we used, except where indicated, were calculated under the following conditions: chromium tube; 26° anode angle; 0.5 mm beryllium window; 60 kV, 50 mA; no filter; incident angle, 57°; emerging angle, 40°. In experiments 8–11, the intensities were also calculated under an emerging angle of 30°. In this way, the number of intensities is doubled, and some extra information is provided. Since our computer program uses the same noise-free calculations, exact solutions are possible: error = 0.

(b) Settings Genetic Algorithm. For the genetic operators as described in the Optimization Using Genetic Algorithms section, we used the settings given below. The number of generations used is fixed at 200 for all experiments, since no significant improvement was observed using more generations. The other settings are also based on prior experience and some preliminary experiments. For more detailed information on this subject, the reader is referred to refs 14–16 and 18.

population size (strings)	100
no. of generations	200
resolution (B) bits	15
elitism ($F_{elitist}$), %	1
threshold ($F_{threshold}$)	20
crossover (P_r), % (uniform)	70
swap rate (P_s), %	50
mutation (P_m), %	2

(c) Computer Program. The computer program was written in ANSI-C and compiled using the GNU C compiler. The genetic algorithm part of the program was created using the genetic algorithm toolbox GATES: Genetic Algorithm Toolbox for Evolutionary Search.[20,21] The program was run on different SUN Sparc stations, all under UNIX. Using this hardware, the maximum running time for complex samples was several hours. For relatively simple samples, the running time varied from roughly 15 to 30 min.

Computer Experiments. The aim of the computer experiments is to see to what extent it is possible to extract qualitative and quantitative structural information from the intensities measured. Therefore, experiments are carried out with varying amounts of structural knowledge. As explained in the Representation section, above, for each layer, a thickness upper estimate must be provided. In all our experiments, we use thickness upper estimates equal to twice the actual thickness, and for nonpresent layers we use 2 μm.

The experiments concern a sample consisting of a 1 μm layer of Ti (20%), As (40%), and Ag (40%) on 30 μm Si (100%) (see Figure 2A) using the following elemental lines: Ti Kα, Ti Kβ, As Kα, As Kβ, As Lα, As Lβ1, Ag Kα, Ag Kβ, Ag Lα, Ag Lβ1, Ag Lβ2, Si Kα, and Si Kβ. In each succeeding experiment, the structural information is reduced by adding nonpresent elemental thicknesses (as defined in the Representation section) to the qualitative sample description. The extra elemental thicknesses represent concentrations of elements that are not present in the corresponding layer and therefore should converge to 0 during optimization. The string representations used in each experiment are shown in Table 1.

(20) Lucasius, C. B.; Kateman, *G. Comput. Chem.* **1994**, *18*, 127–136.
(21) Lucasius, C. B.; Kateman, *G. Comput. Chem.* **1994**, *18*, 137–156.

Table 1. String Representations[a]

expt	representation
1	Ti_1 As_1 Ag_1 Si_2
2	Ti_1 As_1 Ag_1 Si_2 *Ti_2*
3	Ti_1 As_1 Ag_1 *Si_1* Si_2
4	Ti_1 As_1 Ag_1 *Si_1 Ti_2 As_2*
5	Ti_1 As_1 Ag_1 *Si_1* Si_2 *Ti_2*
6	Ti_1 As_1 Ag_1 *Si_1* Si_2 *Ti_2 As_2*
7	Ti_1 As_1 Ag_1 *Si_1* Si_2 *Ti_2 As_2 Ag_2*
8	Ti_1 As_1 Ag_1 *Si_1* Si_2 *Ti_2 As_2 Ag_2*
9	Ti_1 As_1 Ag_1 *Si_1* Si_2 *Ti_2 As_2 Ag_2 Si_3 Ti_3 As_3 Ag_3*
10	Ti_1 As_1 Ag_1 *Si_1* Si_2 *Ti_2 As_2 Ag_2 Si_3 Ti_3 As_3 Ag_3 Si_4 Ti_4 As_4 - Ag_4 Si_5 Ti_5 As_5 Ag_5*

[a] The parameters are all elemental thicknesses. The subscripts denote the layer. Parameters shown in italics are absent in the actual sample.

In experiments 1–7, the number of layers is assumed to be known; elemental thicknesses are added only to present layers.

In experiments 8–10, extra layers that are not present in the sample are also added to the qualitative description. The string representations are depicted in Table 1. The extra elemental thicknesses represent thicknesses of layers and concentrations of elements that are not present in the corresponding layer and therefore should converge to 0 during optimization.

RESULTS

In Tables 2 and 3, experimental results are provided. In these tables, the structural information is presented in terms of concentrations and thicknesses. These concentrations and thicknesses are recalculated from the corresponding elemental thicknesses used in optimization. Each computer experiment is carried

Table 2. Results for a Known Number of Layers[a]

expt	Ti_1 (%)	As_1 (%)	Ag_1 (%)	Si_1 (%)	d_1 (%)	Ti_2 (%)	As_2 (%)	Ag_2 (%)	Si_2 (%)	d_2 (%)	error (%)
comp[b]	20.00	40.00	40.00		1.00				100.00	30.00	
1a	20.00	40.00	40.00	–	1.00	–	–	–	100.00	30.04	1.00×10^{-3}
1b	20.00	40.00	40.00	–	1.00	–	–	–	100.00	30.06	1.01×10^{-3}
2a	19.81	40.05	40.14	–	1.00	0.19	–	–	99.81	48.39	1.50×10^{-1}
2b	19.68	40.13	40.19	–	0.99	0.35	–	–	99.65	31.22	1.51×10^{-1}
3a	20.23	41.11	37.94	0.72	0.99	–	–	–	100.00	12.79	2.98
3b	20.34	41.04	38.63	0.00	0.97	–	–	–	100.00	14.03	2.51
4a	19.74	40.05	40.21	–	0.99	0.27	0.04	–	99.68	24.51	1.61×10^{-1}
4b	19.49	40.24	40.27	–	0.99	0.55	0.00	–	99.45	57.15	2.43×10^{-1}
5a	15.74	41.70	42.57	0.00	0.88	4.34	–	–	95.66	12.11	2.69
5b	10.79	39.27	42.96	6.98	0.84	16.53	–	–	83.47	3.21	5.50
6a	19.75	40.14	40.12	0.00	0.93	0.88	0.65	–	98.46	12.30	1.81
6b	19.60	42.45	37.94	0.00	0.99	0.41	0.00	–	99.59	24.16	3.32
7a	19.81	39.75	39.71	0.72	1.02	0.00	0.00	0.00	100.00	28.20	2.93×10^{-1}
7b	19.70	39.67	39.59	1.04	1.03	0.00	0.00	0.00	100.00	36.50	4.72×10^{-1}

[a] Results obtained after 200 genetic algorithm generations for a sample consisting of a 1 μm Ti (20%), As (40%), and Ag (40%) layer on 30 μm Si (100%). Each row consists of results obtained in one genetic algorithm run. The letters a and b are used to distinguish between duplicate runs. Duplicates are obtained by initiating the genetic algorithm run with a different random generator seed. Each column represents the results for the corresponding parameter. The subscripts denote the layer for which the parameter is optimized. A – denotes that the parameter was not included in the optimization. [b] The actual sample composition.

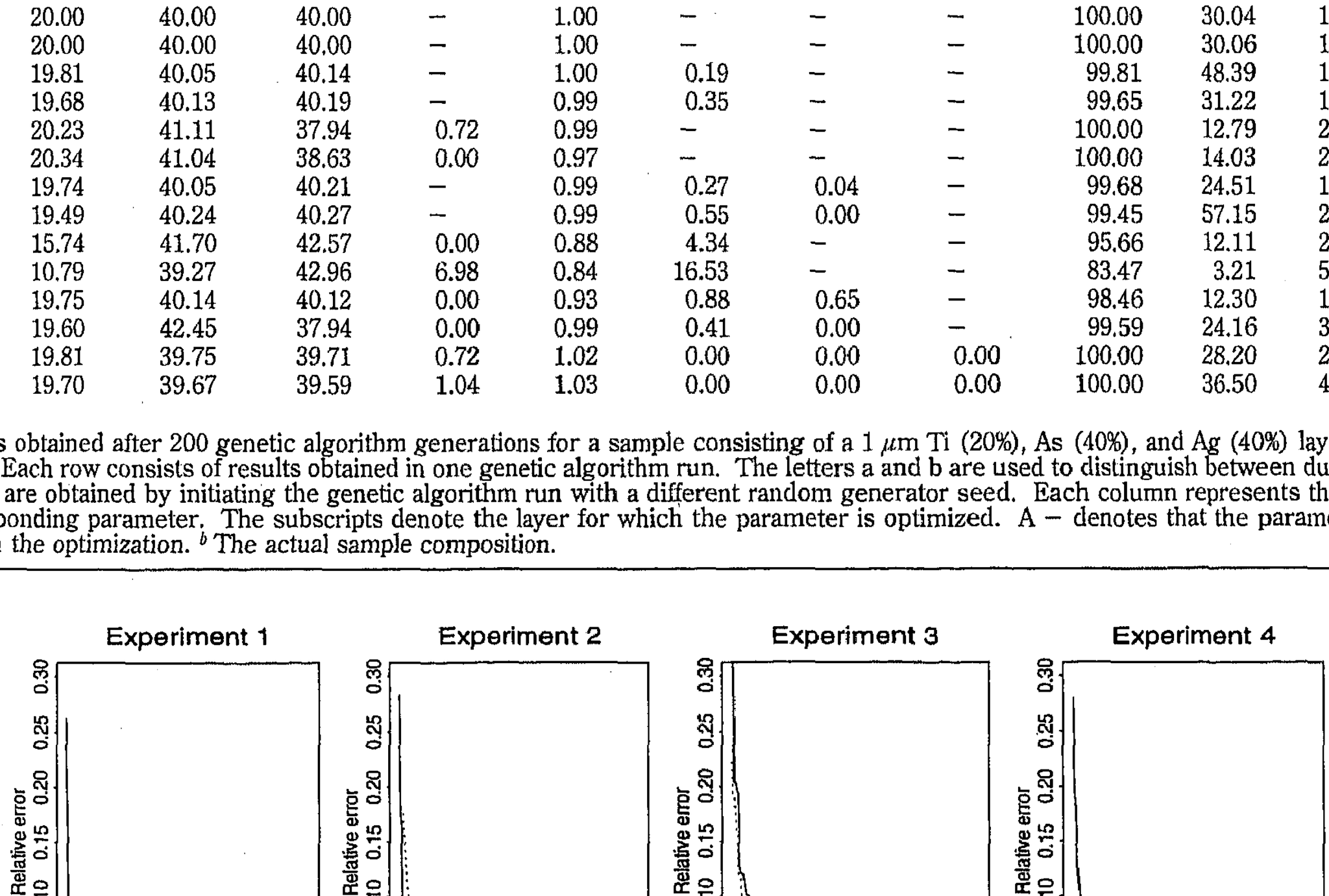

Figure 3. Elementwise relative error as a function of the number of generations. –, Run 1; ···, run 2.

Table 3. Results for an Unknown Number of Layers[a]

		expt					
parameter	comp[b]	8a	8b	9a	9b	10a	10b
Ti_1 (%)	20.0	19.94	19.62	18.82	16.99	18.47	13.86
As_1 (%)	40.0	39.97	40.15	40.65	40.73	40.30	43.37
Ag_1 (%)	40.0	39.95	40.19	40.53	42.27	41.05	42.75
Si_1 (%)		0.13	0.04	0.00	0.00	0.18	0.02
d_1 (μm)	1.0	1.01	0.99	0.97	0.90	0.96	0.88
Ti_2 (%)		0.00	0.39	1.30	4.17	1.69	6.05
As_2 (%)		0.00	0.00	0.02	0.39	0.01	0.03
Ag_2 (%)		0.00	0.00	0.02	0.08	0.11	0.01
Si_2 (%)	100.0	99.99	99.61	98.66	95.36	98.19	93.91
d_2 (μm)	30.0	37.53	58.27	28.03	10.20	14.74	14.29
Ti_3 (%)		–	–	2.47	0.41	11.76	17.57
As_3 (%)		–	–	0.18	8.32	1.15	2.04
Ag_3 (%)		–	–	0.94	4.89	1.50	11.95
Si_3 (%)		–	–	96.41	86.38	85.59	68.42
d_3 (μm)		–	–	0.36	0.54	0.49	0.19
Ti_4 (%)		–	–	–	–	11.87	18.61
As_4 (%)		–	–	–	–	1.28	0.60
Ag_4 (%)		–	–	–	–	0.88	0.13
Si_4 (%)		–	–	–	–	85.97	80.65
d_4 (μm)		–	–	–	–	0.24	0.44
Ti_5 (%)		–	–	–	–	4.06	23.86
As_5 (%)		–	–	–	–	3.40	1.39
Ag_5 (%)		–	–	–	–	0.55	1.43
Si_5 (%)		–	–	–	–	91.99	73.33
d_5 (μm)		–	–	–	–	0.53	0.65
Ti_6 (%)		–	–	–	–	3.26	10.98
As_6 (%)		–	–	–	–	1.47	0.00
Ag_6 (%)		–	–	–	–	1.12	0.14
Si_6 (%)		–	–	–	–	94.14	88.89
d_6 (μm)		–	–	–	–	0.41	0.47
error (%)		0.1	0.2	0.5	1.9	1.4	3.1

[a] Results obtained after 200 genetic algorithm generations for a sample consisting of a 1 μm Ti (20%), As (40%), and Ag (40%) layer on 30 μm Si (100%). Each row consists of results obtained in one genetic algorithm run. The letters a and b are used to distinguish between duplicate runs. Duplicates are obtained by initiating the genetic algorithm run with a different random generator seed. Each column represents the results for the corresponding parameter. The subscripts denote the layer for which the parameter is optimized. A – denotes that the parameter was not included in the optimization. The problem complexity in the experiments is increased by adding extra layers to the intial estimate which the are not present in the sample. [b] The actual sample composition.

out in duplicate by initiating the genetic algorithm run with two different random seeds. In the tables, duplicate experiments are indicated with the letters a and b. A – denotes that the parameter was not present on the string. The actual sample composition is denoted by comp.

Known Number of Layers. The results for the Ti,As,Ag/Si sample, with representations in which the number of layers is known, are shown in Table 2. The final sample structures found all clearly reflect the quantitative solution. For all computer experiments except experiment 5b, errors are smaller than 5%. From this observation, one can conclude that the representation and the settings of the genetic algorithm are good. The exact solutions are not obtained, but the low search precision is a property of genetic algorithms (see the Optimization Using Genetic Algorithms section). One of the most striking results is the prediction of the thickness of the second layer (d_2). In experiments 3, 5, and 6, this thickness is much too low, and in experiments 2a, 4b, and 7b, d_2 is estimated too high. Compared with the overestimation of the second layer thickness, the effect on the final error of underestimating this thickness is much larger. The information for calculating the thickness of the second layer is in the layer on top of it and in the silicon fluorescence. In all computer experiments, the upper layers are almost the same, so the differences must be explained by the nature of the silicon fluorescence. Radiation which originates deeper than the escape depth of silicon Kα radiation does not leave the sample because it is attenuated by the material on top of it. That is why a thicker layer does not contribute extra to the intensities and so does not affect the error. On the other hand, if the layer thickness is smaller than the escape depth of silicon Kα radiation, then the silicon intensities are too low, causing a larger error.

In experiment 5, not only d_2 but also titanium is predicted very badly. In the final solution, much of the titanium is located in the second layer. This seems to be compensated by a smaller d_2 in which the titanium content is too low.

A possible explanation for both the badly predicted titanium concentrations and the thicknesses of the second layer is the fact that silicon (d_2) and titanium are both relatively light elements. The radiation for these elements is not as energetic as the radiations for the heavier elements. This could be the reason why their prediction it is more difficult.

Both difficulties in predicting Ti and Si (d_2) seem to have a noticeable effect on the final error.

Figure 3 depicts the best solution present in the population as a function of the number of generations. In all experiments, the largest decrease in error takes place during the first generations. The neighborhood of the global optimum is approached, but the exact location is not obtained.

Unknown Number of Layers. The results for the Ti,As,-Ag/Si sample with representations in which the number of layers is not known are shown in Table 3. In all cases, the quantitative sample structure is described reasonably by the first two layers. In the deeper layers, silicon is the main element. If these relatively thin layers are ignored or combined with the second layer, then the results and conclusions are almost the same as those for experiments 1–7.

CONCLUSIONS

This paper presents a preliminary genetic algorithm for model-free X-ray fluorescence analysis of thin and multiple-layer films. The experiments described concern simulated data for which perfect solutions exist. These solutions are characterized by a spectral error equal to 0 and an exact quantitative sample composition. Results are presented for a sample consisting of two layers. Analyses of more complex samples were also attempted but were not successful. This is a subject for future investigation.

The first experiments use input models that assume an unknown distribution of the elements over the layers. The method and these estimates applied to relatively simple two-layer samples yield spectral errors smaller than 5% and good quantitative sample structures. When the input models assume more layers than present, the quantitative sample structure in the final solution is described reasonably by the upper layers. However, it is difficult to decide how many layers are actually present. We hope to overcome this problem by making use of the contributions of the individual layers to the intensities calculated.

Perfect solutions are not obtained since genetic algorithms can very effectively locate the neighborhood of the global solution, but they are very bad at locating the exact position. For obtaining exact solutions, the best solution of the genetic algorithm run could be used as the starting point of a steepest descent method. We are working on this subject at the moment. Once this

combination is made, it can be concluded if the solutions obtained in genetic algorithm optimization are good enough to recalculate the actual composition. If good solutions are obtained, we plan to test our method on real-world data.

SUPPORTING INFORMATION AVAILABLE

Tables listing the predicted intensities obtained in experiments 1–7 and the simulated intensities and predicted intensities obtained in experiments 8–10 (2 pages). Ordering information is given on any current masthead page.

Received for review November 28, 1995. Accepted April 23, 1996.[⊗]

AC951152T

[⊗] Abstract published in *Advance ACS Abstracts,* June 1, 1996.